NED UNIVERSITY
OF ENGINEERING AND TECHNOLOGY

Course/Course Code:

Data Acquisition and Microcontroller (MC-546)

Lecture # 08

Course Teacher:

Dr. Tariq Rehman

Software for Data Acquisition Systems

- ☐ The design of data acquisition (DAQ) and control software must have the capacity to recover gracefully from instrument component failures and power outages without losing data.
- ☐ Data Acquisition software must be easily reconfigurable and provides a high-level language for algorithm design. Moreover, it is required a data archiving method that can verify the integrity of the data acquired.
- ☐ In the software design, it is possible to confront with two points:
  1. **For customer:** a user-friendly software.
  2. **For supplier:** a powerful mechanism which can reflect all the combination of hardware facilities.
- ☐ Good design is a negotiation of a process that tries to reconcile the two points, working top-down from the client side and bottom-up from the supplier side.

Software for Data Acquisition Systems

- ☐ The process of software design (Fig.1) consists of developing intermediate algorithms of abstraction: a set of efficiently implementable algorithms.
- ☐ Software package usually has analysis and presentation capabilities into the DAQ software.
- ☐ Your application software normally does such tasks as:
  1. Real-time monitoring
  2. Data analysis
  3. Data logging
  4. Control algorithms
  5. Human machine interface (HMI).

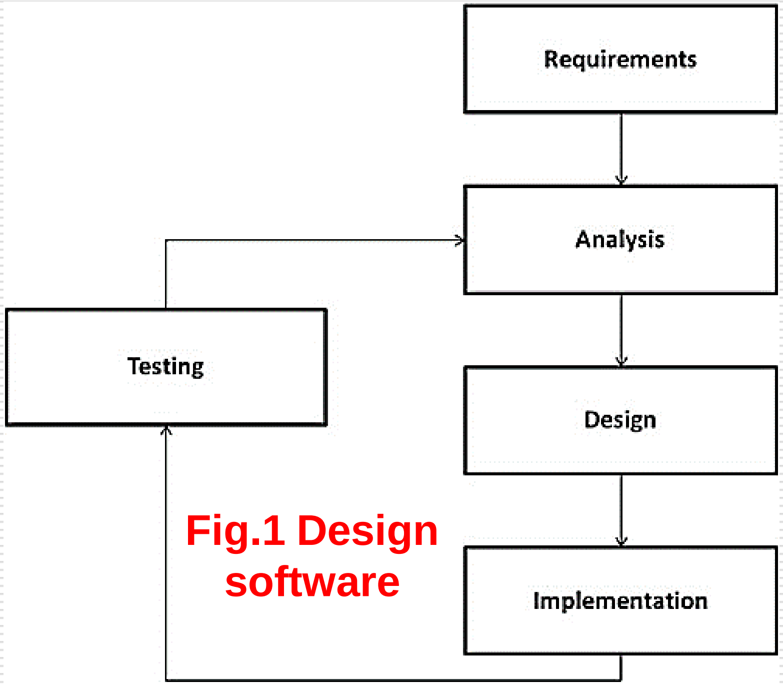

Fig.1 Design software

LabView Software

- LabView is a graphical programming language from National Instruments.
- It includes the Front Panel and the Block Diagram.
- **Front Panel** is like User Interface, can be composed of controls inputs, shows outputs, and connects to the engine.
- **Block Diagram**, instead, allows it to function and connects everything together.
- Some advantages can be described in the following points:
    1. Good for building piecewise: using small amounts of code in a larger code
    2. Visually programming is easier to learn
    3. NI has many pieces of hardware that are easily compatible with LabVIEW, but also can be connected to any hardware

Android for DAQ

- Android is a Linux-based operating system designed primarily for smartphones and tablet computers.
- It is currently developed by Google in conjunction with the Open Handset Alliance.
- Android consists of a kernel based on the Linux kernel 2.6 and Linux Kernel 3.x (Android 4.0 onwards), with libraries and APIs written in C language and application software running on a framework which includes Java-compatible libraries based on Apache Harmony.
- The main hardware platform for Android is the ARM architecture. It supports for x86 from the Android x86 project.
- Personal computer gave engineers the ability to create smaller and more portable DAQ systems using C-language,
- In this regard, are the tablet computers, a natural evolution of this trend?

Android for DAQ

- ☐ **Tablets** work under different operating systems and have less processing power than PC.
- ☐ There are two points to consider:
  1. **C and Labview** are **not supported** in Android or IOS;
  2. **USB**, **a common DAQ bus,** is found in a set of Tablets, in other case, an adapter is required.
- ☐ In these instances, moving a possible data acquisition system to a tablet requires extra attention.
- ☐ The tablet offers development environment for DAQ application. There are three main considerations in creating a DAQ application on a tablet:
  1. **Hardware connectivity:** the Tablet has few options of control, such as Wi-fi and bluetooth.
  2. **Program language support:** Tablets support Android done in Java.
  3. **Device driver availability:** device drivers permits a high-level mode to easily and reliably execute DAQ board functionality.

Design of Firmware

- **Firmware** is a software program or set of instructions programmed on a hardware, installed on ROM chips (ROM, PROM, EPROM) or flash chips.
- It enables the device to render its capabilities functionally. Moreover, it coordinates the activities of the hardware during normal operation and contains programming constructs used to perform those operations.
- Example: in a typical modem, the firmware will be a factor in establishing the modem’s data rate, command set recognition, and special feature implementation.
- The node firmware (Fig. 2) is being formed from three entities:
    1. Data structures that hold state information, behavior information, and map information.
    2. Background processes that handle actions implemented in hardware or using hardware peripherals such as the wireless communication and the sensor readings.
    3. The main firmware code that executes the node’s activity.

Design of Firmware

- The **flowchart** in Fig. 2 describes the basic operations if the node is in the idle state.
- The node can also be in the attached and the sensing state, respectively.

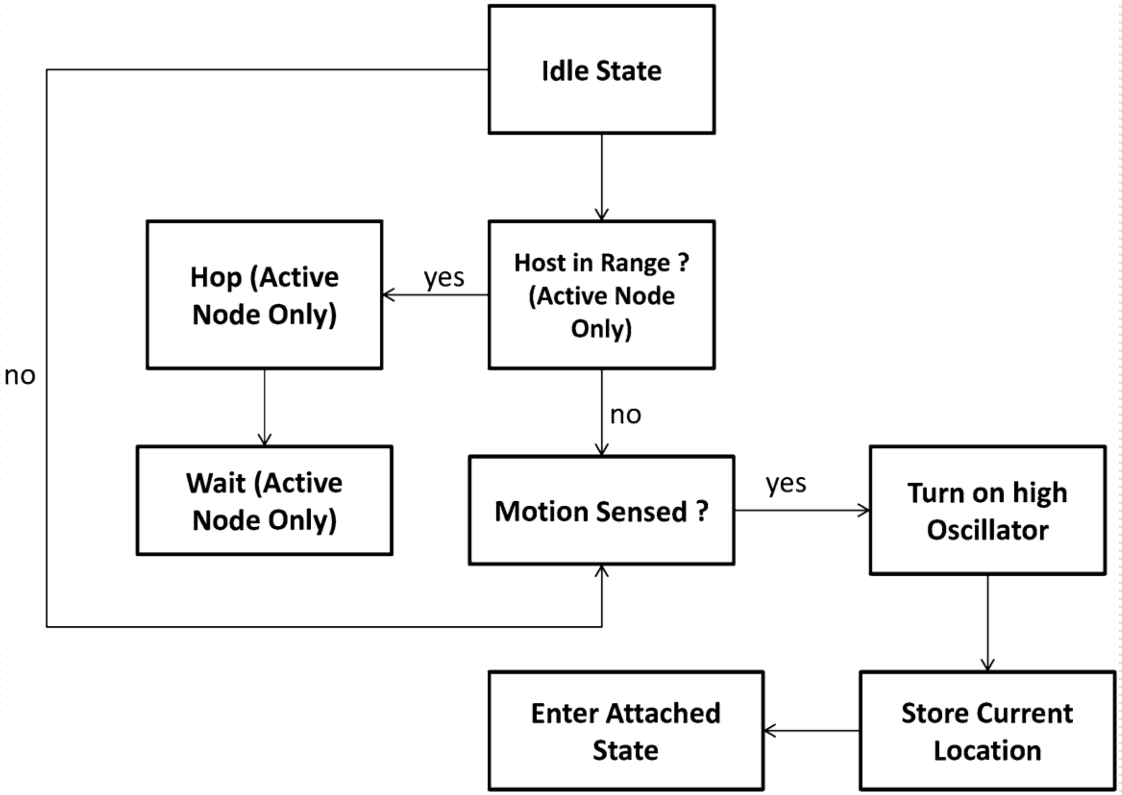

Fig. 2 Example of flow diagram of firmware

Smart Data Acquisition System (MAX1329/MAX1330)

- ☐ The **MAX1329/MAX1330** are smart data acquisition systems (DASs or DAQs) based on a successive approximation register (SAR) analog-to-digital converter (ADC).
- ☐ These devices are highly integrated, offering an ADC, digital-to-analog converters (DACs), operational amplifiers (op amps), voltage reference, temperature sensors, and analog switches in the same device.
- ☐ The MAX1329 (Figs. 3 and 4) offers a single ADC with a reference buffer.
- ☐ The ADC is capable of operating in one of two user-programmable modes.

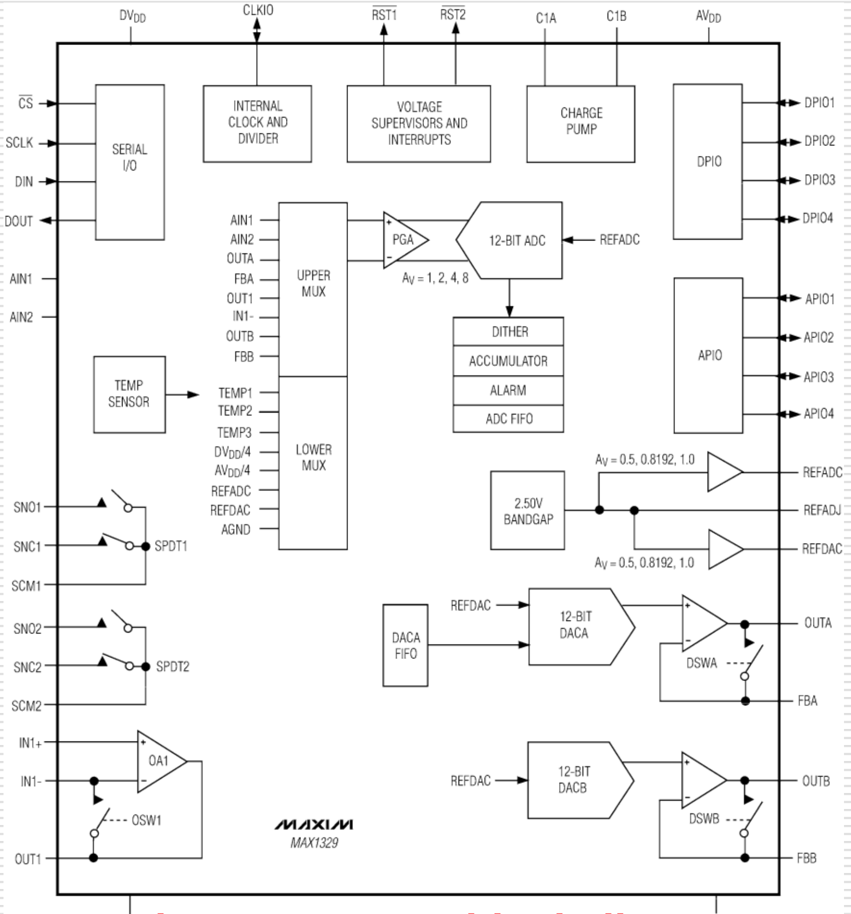

Fig. 3 MAX1329, block diagram

MAX1329/MAX1330

- In **normal mode**, the ADC provides up to 12 bits of resolution at 312 ksps. In **DSP mode**, the ADC provides up to 16 bits of resolution at 1,000 sps.
- The ADC accepts one external differential input or two external single-ended inputs as well as inputs from other circuitry on-board.
- An on-chip programmable gain amplifier (PGA) follows the analog inputs, reducing external circuitry requirements. The PGA gain is adjustable from 1 to 8 V/V.
- The MAX1329 operate from a 1.8 to 3.6 V digital power supply.

*ksps= kilosample/second

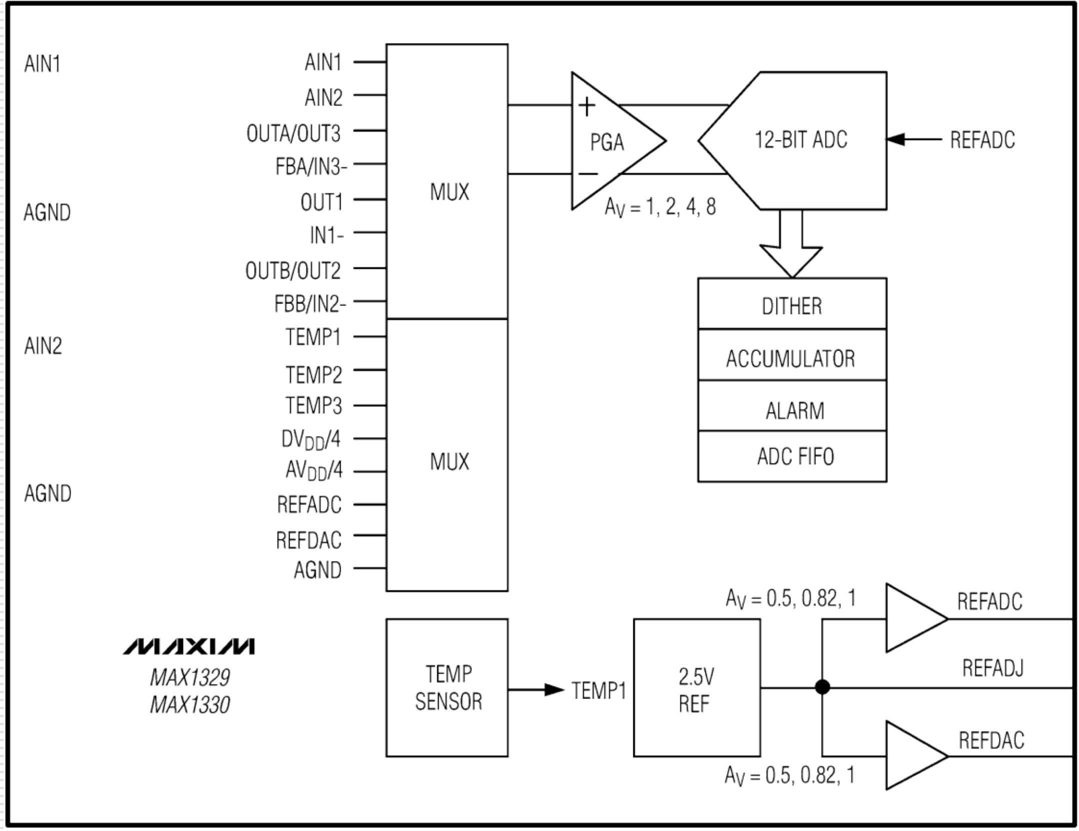

Fig. 4 MAX1329, block diagram

MAX1329/MAX1330

- Shutdown and sleep modes are available for power-saving applications. Under normal operation, an internal charge pump boosts the supply voltage for the analog circuitry when the supply is <2.7 V.
- The MAX1329 offer **four** analog programmable I/Os (APIOs) and **four** digital programmable I/Os (DPIOs).
- The **APIOs** can be configured as general-purpose logic inputs and outputs, as a wakeup function, or as a buffer and level shifter for the serial interface **to communicate with slave devices** powered by the analog supply, AVDD.
- The **DPIOs** can be configured as general-purpose logic inputs and outputs as well as inputs to directly control the ADC conversion rate, the analog switches, the loading of the DACs, wake-up, sleep, and shutdown modes, and as an interrupt when the analog-to-digital conversion is complete.

MAX1329/MAX1330

- ☐ The MAX1329 includes dual 12-bit force-sense DACs with a programmable reference buffer and one op amp. For the MAX1329, a 16-word DAC FIFO can be used with the DACA for direct digital synthesis (DDS) of waveforms.

Features:

1. 1.8–3.6 V single digital supply operation
2. Internal charge pump for analog circuits (2.7–5.5 V)
3. 12-Bit SAR ADC; 12 bits, 312 ksps, no missing codes; 16 bits, 1,000 sps, DSP mode; 16-word FIFO and 20-bit accumulator
4. PGAwithgainsof1,2,4,and8
5. Unipolar and bipolar modes
6. 16-Input differential multiplexer

MAX1329/MAX1330

7. Adjustable reference buffers provide 1.25, 2.048, or 2.5 V
8. ADC alarm register
9. Uncommitted op amps
10. Dual SPDT analog switches
11. Internal/external temperature sensor
12. Internal oscillator with clock I/O (Fig. 6.6)
13. Digital programmable I/O
14. Analog programmable I/O
15. Programmable interrupts
16. Accurate supply voltage measurement
17. Programmable dual voltage monitors

Circuit Application

- The MAX1329 (Figs. 5 and 6) features a 4-wire serial interface consisting of a **chip select (CS)**, **serial clock (SCLK), data in (DIN),** and **data out (DOUT).**
- The data are clocked in at **DIN** into the shift register on the **rising edge of SCLK**.
- Data are clocked out at **DOUT** on the **falling edge of SCLK**.
- The serial interface is compatible with SPI modes.
- The **MAX1329 performs temperature measurement** by measuring the voltage across a diode-connected transistor at two different current levels.
- The temperature measurement process is fully automated in the MAX1329.
- All steps are sequenced and executed by the MAX1329 each time an input channel (or an input channel pair) configured for temperature measurement is scanned.

Circuit Application

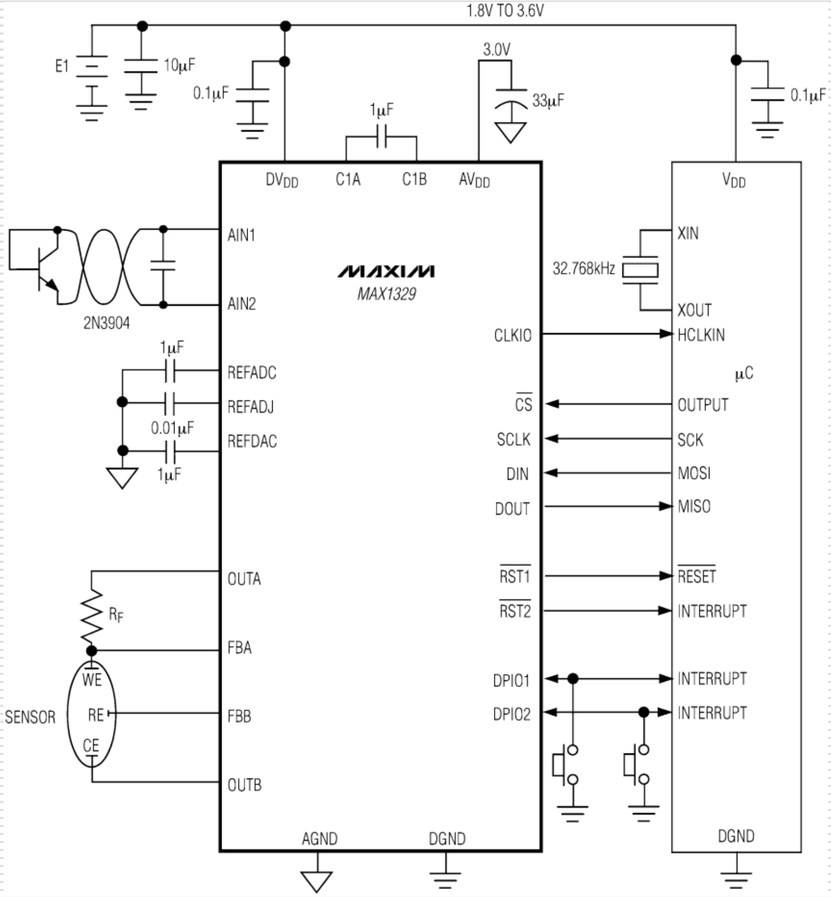

Fig 5 Circuit application

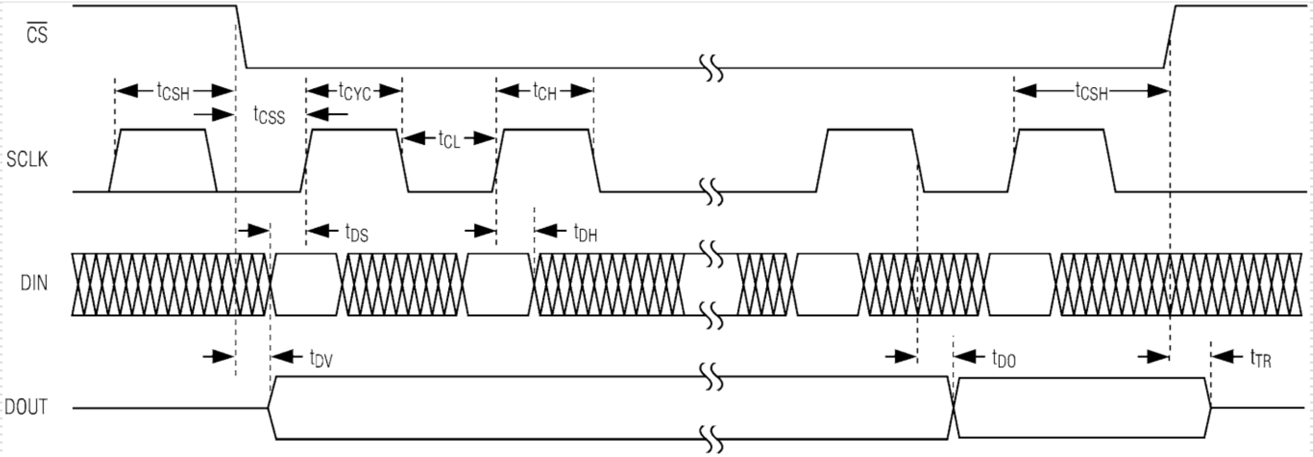

Fig. 6 Detailed serial-interface timing diagram [1]

Complete DAQ

- A complete DAQ using MAX1329 is visualized in Fig. 7.
- A bus interface (e.g., USB) needs to manage the data with several buses.
- MAX3420E circuitry can be necessary to implement a full-speed USB peripheral.
- This DAQ can be used to acquire data from sensors: temperature, position sensor, and so on.
- In the project is included also **slot expander** to increase the functionality of the device: for example **bluetooth module, VME module, GPS module**, and so on.

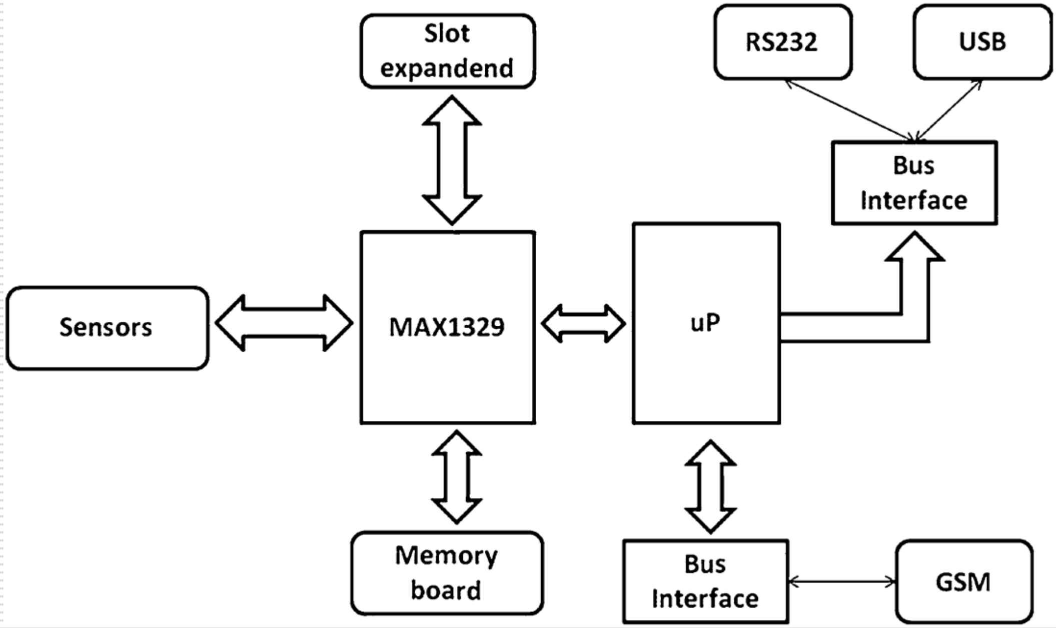

Fig. 7 Block diagram DAQ

END of Lecture # 08